

IN THE CLAIMS

1. (Currently Amended) A method for providing redundant data load sharing in a distributed network, comprising:

receiving an original entry one or more original data entries at a respective one of a plurality of nodes;

storing the each original data entry in a first one of a in its respective one of the plurality of nodes;

generating a replicated data entry from the each original data entry at its respective one of the plurality of nodes;

storing the each replicated data entry in a second one of the plurality of nodes node different from where its associated original data entry is stored, each original data entry including information as to a location of its associated replicated data entry, each replicated data entry including information as to a location of its associated original data entry;

identifying at a particular one of the plurality of nodes a failure of either of the first one or the second one another one of the plurality of nodes;

determining whether the particular one of the plurality of nodes includes an original data entry or a replicated data entry having information as to the location of a replicated data entry or an original data entry respectively that matches an identified failed node;

generating a re-replicated data entry in the non-failed first or second particular one of the plurality of nodes for storage at a third separate working one of the plurality of nodes in response to the failure in the first or second one of the plurality of nodes information matching the identified failed node.

2. (Currently Amended) The method of Claim 1, further comprising:

determining whether there is sufficient capacity in the distributed network to handle data entry replication in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node.

3. (Currently Amended) The method of Claim 2, further comprising:

preventing replication of the original or replicated data entry at the ~~third~~ separate working one of the plurality of nodes in response to insufficient capacity in the distributed network.

4. (Currently Amended) The method of Claim 3, further comprising:

dynamically adjusting the capacity of the distributed network in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node in order to store new original data entries without replication.

5. (Currently Amended) The method of Claim 3, further comprising:

identifying a recovery of the identified failed ~~first or second one of the plurality of nodes~~ node;

performing storage and replication of subsequently received original data entries in response to the recovery of the identified failed ~~first or second one of the plurality of nodes~~ node.

6. (Currently Amended) The method of Claim 5, further comprising:

adjusting the capacity of the distributed network in response to the recovery of the identified failed ~~first or second one of the plurality of nodes~~ node.

7. (Currently Amended) The method of Claim 5, further comprising:

performing replication of those data entries previously stored but not replicated as a result of the ~~failure of the first or second one of the plurality of nodes~~ recovery of the identified failed node.

8. (Currently Amended) The method of Claim 1, further comprising:

identifying a recovery of the identified failed ~~first or second one of the plurality of nodes~~ node;

including the recovered identified failed ~~first or second one of the plurality of nodes~~ node in the storage and replication of subsequent original data entries.

9. (Original) The method of Claim 1, further comprising:
establishing a capacity for the distributed network, the capacity representing an amount of data to be stored in the distributed network;

establishing a minimum number of the plurality of nodes required to provide redundancy in the distributed network.

10. (Original) The method of Claim 9, wherein at least a single occurrence of all data entries are maintained in the plurality of nodes when the number of plurality of nodes falls to one less than the minimum number.

11. (Currently Amended) A system for providing redundant data load sharing in a distributed network, comprising:

a plurality of nodes, ~~a first one of the plurality of nodes~~ operable to receive and store two copies of a data entry, wherein the two copies of a data entry are stored in separate nodes, each data entry including information as to a location of its associated data entry copy, an original data entry, the first one of the plurality of nodes operable to provide a replicate data entry to a second one of the plurality of nodes, the first one each of the plurality of nodes operable to ~~determine~~ identify a failure of the ~~second one of the plurality of nodes, any node, each node operable to determine whether it includes a data entry having information as to the location of its associated data entry copy that matches an identified failed node, each node~~ ~~the first one of the plurality of nodes~~ operable to generate provide a re-replicated data entry for storage in a separate working to a third one of the plurality of nodes in response to failure of the second one of the plurality of nodes having a data entry with information that matches an identified failed node.

12. (Currently Amended) The system of Claim 11, wherein each node includes a distributed control function operable to control storage and replication of ~~the original data entry entries.~~

13. (Currently Amended) The system of Claim 11, wherein ~~the first one of the plurality of nodes~~ each node is operable to determine whether there is sufficient capacity in the distributed network to handle data entry replication in response to the ~~failure of the second one of the plurality of nodes~~ identified failed node.

14. (Currently Amended) The system of Claim 13, wherein ~~the first one of the plurality of nodes~~ each node is operable to prevent replication ~~of the original~~ of a data entry ~~at the third~~ for storage at a separate working one of the plurality of nodes in response to insufficient capacity in the distributed network.

15. (Currently Amended) The system of Claim 14, wherein the plurality of nodes are operable to dynamically adjust the capacity of the distributed network in response to the ~~failure of the second one of the plurality of nodes~~ identified failed node in order to store new data entries without replication.

16. (Previously Presented) The system of Claim 11, wherein the capacity of each of the plurality of nodes is adjusted in response to an addition of a new node or failure of an existing node.

17. (Currently Amended) A system for providing redundant data load sharing in a distributed network, comprising:

means for receiving ~~an original entry~~ one or more original data entries at a respective one of a plurality of nodes;

means for storing ~~the~~ each original data entry ~~in a first one of a~~ in its respective one of the plurality of nodes;

means for generating a replicated data entry from ~~the~~ each original data entry at its respective one of the plurality of nodes;

means for storing ~~the~~ each replicated data entry in a ~~second one of the plurality of nodes~~ node different from where its associated original data entry is stored, each original data entry including information as to a location of its associated replicated data entry, each replicated data entry including information as to a location of its associated original data entry;

means for identifying at a particular one of the plurality of nodes a failure of ~~either of the first one or the second one~~ another one of the plurality of nodes;

means for determining whether the particular one of the plurality of nodes includes an original data entry or a replicated data entry having information as to the location of a replicated data entry or an original data entry respectively that matches an identified failed node;

means for generating a re-replicated data entry in the ~~non-failed first or second~~ particular one of the plurality of nodes for storage at a ~~third~~ separate working one of the plurality of nodes in response to the ~~failure in the first or second one of the plurality of nodes~~ information matching the identified failed node.

18. (Currently Amended) The system of Claim 17, further comprising:

means for determining whether there is sufficient capacity in the distributed network to handle data entry replication in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node.

19. (Currently Amended) The system of Claim 18, further comprising:

means for preventing replication of the original or replicated data entry at the ~~third~~ separate working one of the plurality of nodes in response to insufficient capacity in the distributed network.

20. (Currently Amended) The system of Claim 19, further comprising:

means for dynamically adjusting the capacity of the distributed network in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node in order to store new original data entries without replication.

21. (Currently Amended) The system of Claim 19, further comprising:

means for identifying a recovery of the identified failed ~~first or second one of the plurality of nodes~~ node;

means for performing storage and replication of subsequently received original data entries in response to the recovery of the identified failed ~~first or second one of the plurality of nodes~~ node.

22. (Currently Amended) A computer readable medium including code for providing redundant data load sharing in a distributed network, the code operable to:

receive an original entry one or more original data entries at a respective one of a plurality of nodes;

store the each original data entry in a first one of a in its respective one of the plurality of nodes;

generate a replicated data entry from the each original data entry at its respective one of the plurality of nodes;

store the each replicated data entry in a second one of the plurality of nodes node different from where its associated original data entry is stored, each original data entry including information as to a location of its associated replicated data entry, each replicated data entry including information as to a location of its associated original data entry;

identify at a particular one of the plurality of nodes a failure of either of the first one or the second one another one of the plurality of nodes;

determine whether the particular one of the plurality of nodes includes an original data entry or a replicated data entry having information as to the location of a replicated data entry or an original data entry respectively that matches an identified failed node;

generate a re-replicated data entry in the non-failed first or second particular one of the plurality of nodes for storage at a third separate working one of the plurality of nodes in response to the failure in the first or second one of the plurality of nodes information matching the identified failed node.

23. (Currently Amended) The computer readable medium of Claim 22, wherein the code is further operable to:

determine whether there is sufficient capacity in the distributed network to handle data entry replication in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node.

24. (Currently Amended) The computer readable medium of Claim 23, wherein the code is further operable to:

prevent replication of the original or replicated data entry at the ~~third~~ separate working one of the plurality of nodes in response to insufficient capacity in the distributed network.

25. (Currently Amended) The computer readable medium of Claim 24, wherein the code is further operable to:

dynamically adjust the capacity of the distributed network in response to the ~~failure of the first or second one of the plurality of nodes~~ identified failed node in order to store new original data entries without replication.

26. (Currently Amended) The computer readable medium of Claim 24, wherein the code is further operable to:

identify a recovery of the identified failed ~~first or second one of the plurality of nodes~~ node;

perform storage and replication of subsequently received original data entries in response to the recovery of the identified failed ~~first or second one of the plurality of nodes~~ node.